

REMARKS

In reply to the Office Action of December 15, 2003, Applicant submits the following remarks. Claims 1-11, 16-18, 23-25, 34-35, 40-41 and 44-45 have been amended. Claims 46-54 are new. No new matter has been added. Claims 1-54 are now pending after entry of this amendment. Applicant respectfully requests reconsideration in view of the foregoing amendments and these remarks.

I. Section 102 Rejections

Claims 2-5, 8, 16-18, 21-28, 30-35 and 40-41 were rejected under 35 U.S.C. 102(b) as allegedly being anticipated by U.S. Patent Number 6,351,775 ("Yu"). Applicant respectfully disagrees.

Claims 2-5 and 8

Amended claim 2 recites a method of centrally managing distributed components. The method includes storing in a first computer system a central registry database including configuration information related to distributed components. The distributed components are located in remote computer systems. The first computer system and the distributed components are in a distributed enterprise application system. Requests are received from the distributed components in an enterprise application system for configuration information updates. Each of the distributed components communicates with one or more enterprise applications. Configuration changes to be implemented are determined in response to the requests. The central registry database is modified to reflect at least a portion of the configuration changes. The configuration changes are allocated to the corresponding distributed components. The configuration changes are transferred to and implemented in the corresponding distributed components.

Yu describes dynamic routing of object requests among a collection of servers such as proxy servers and Web servers in the Internet environment (col. 1, lines 22-29). Requestors 110-153 request services from a server cluster of a network 105 (Fig. 1, col. 5, lines 56-67). The network can be the Internet, the World Wide Web, an Intranet and local area networks (*id.*).

Typical requests for objects include World Wide Web pages accesses, remote file transfers, electronic mail and transaction support (col. 6, lines 7-9). The server cluster includes server nodes 161-163 to handle high traffic demand (col. 5, line 67 and col. 6, lines 1-2). The requests can be processed by any of the server nodes, however routing requests for the same object to a single server node will result in a better cache hit probability (col. 6, lines 10-17). An arbitrator 235 monitors the load of each server and updates the load balancing among the servers (FIG. 2, col. 6, lines 31-36, col. 7, lines 58-67). When the requestor sends a request for an object, the requestor sends the request to a server specified in a class-to-server assignment table (col. 11, lines 35-44). If the server is not available for fulfilling the request, a mapping request is sent to the arbitrator (*id.*). The arbitrator determines the mapping from the arbitrator's assignment table (col. 10, lines 66-67 and col. 11, lines 1-35). The new class-to-server assignment is sent to the requestor (col. 11, lines 11-35). The requestor updates the local assignment table to reflect the new mapping (*id.*). The arbitrator can update the mapping table by sending statistics collection requests to the servers and running a reassignment routine (col. 9, lines 51-67 and col. 10, lines 1-24). An update request is communicated to the servers with the updated assignment table (col. 9, lines 44-50).

Yu describes a method of updating a requestor's class-to-server assignment table. Yu's method is directed at allocating the load of object requests amongst a cluster of servers that handle traffic on the World Wide Web. The requestors and arbitrators described in Yu are not necessarily related to one another other than by the Internet. Yu's requestors are not components in an enterprise application system and do not communicate with one or more enterprise applications. Yu's servers are not components in an enterprise applications system. Further, while Yu's servers may communicate with Yu's requestors, Yu's servers do not generate requests that contain configuration information updates. As described above, Yu's arbitrator sends updates to the servers. Yu's servers do not submit configuration information updates to Yu's arbitrator. Rather, the servers submit statistical information after the request for such statistics has been made. The Applicant's claimed invention requires that requests are received from distributed components in an enterprise application system for configuration information

updates, that the distributed components communicates with one or more enterprise applications and that configuration changes are transferred to and implemented in the corresponding distributed components. For at least this reason, Yu fails to anticipate the limitations of claim 2.

Claim 3 depends from claim 2 and requires that the configuration information includes at least one of data translation, routing, formatting, scheduling, collaborations and message identification. Yu does not teach or suggest configuration information related to any of data translation, routing, formatting, scheduling, collaborations or message identification. In addition to the reasons provided above with respect to claim 2, for at least this reason Applicant submits that Yu does not anticipate claim 3.

Claim 5 depends from claim 2 and requires that the central registry database communicates with a plurality of subordinate registry databases in the first computer system, and the plurality of subordinate registry databases are in communication with the distributed components. Yu does not teach or suggest a subordinate registry database. For at least this reason and those provided with respect to claim 2, Applicant submits that Yu does not anticipate claim 5.

Claim 8 depends from claim 2 and requires that the central registry database communicates with a plurality of duplicate registry databases in the first computer system, and the plurality of duplicate registry databases are in communication with the distributed components. Yu does not suggest or disclose duplicate registry databases in a first computer system, much less that a plurality of duplicate registry databases are in communication with distributed components. In addition to the reasons provided above with respect to claim 2, for at least this reason Applicant submits that Yu does not anticipate claim 8.

Claims 16-18 and 21-22

Amended claim 16 recites a method of centrally managing distributed components. The database is stored on a first computer system and the distributed components are stored in remote computer systems. The distributed components communicate with a enterprise applications. Requests are received from one of the distributed components for configuration update requests.

The configuration changes are determined and the central registry database is modified to reflect at least a portion of the changes. The changes are transferred to a distributed component.

Yu describes an arbitrator that sends out statistics collection requests to servers. The arbitrator executes the statistics and evaluation routine. Yu does not teach or suggest receiving requests from a distributed component, determining configuration changes in response to the requests and transferring the configuration changes to a distributed component, as required by amended claim 16. Rather, Yu describes an arbitrator that sends requests for statistics and servers that receive update requests. Yu also describes requestors that send out mapping requests to the arbitrators. The arbitrators receive the mapping requests from requestors. As described above, the requestors are not components of an enterprise application. Thus, Yu does not teach or suggest a method including storing a database in a first computer system that contains configuration information related to distributed components that are in remote computer systems and communicate with enterprise applications as required by amended claim 16. For at least these reasons, claim 16 is not anticipated by Yu. Claim 17-18 and 21-22 depend from claim 16 and Applicant submits the claims are allowable for at least the same reasons.

Claim 23 recites a distributed enterprise application integration system having a central control module and a plurality of distributed components. The central control module includes a central registry database used to store configuration data about a distributed enterprise application system. The distributed components are configured to communicate with one or more enterprise applications.

Yu does not suggest or disclose a distributed enterprise application integration system having a control module and distributed components where the distributed components are configured to communicate with one or more enterprise applications. Further, Yu does not suggest or disclose a central registry database used to store configuration data about a distributed enterprise application system. Rather, Yu discloses an arbitrator that stores an assignment table for class-to-server mapping where the servers deliver services to any requestor that accesses the server over the Internet. For at least these reasons, Applicant submits that claim 23 is not anticipated by Yu. Claims 24-27 depend from claim 23 and are similarly not anticipated.

Claim 28

Claim 28 recites a distributed, multi-platform application integration system including a plurality of multi-platform applications.

Yu does not suggest or disclose a system that includes a plurality of multi-platform applications. For at least this reason, Applicant submits that claim 28 is not anticipated by Yu. Claims 30-35 depend directly from claim 28 and are not anticipated by Yu for at least the same reason.

Claim 40

Claim 40 recites a method for integrating distributed applications. The method includes collecting configuration change information from a plurality of distributed components related to requests for configuration changes. The configuration changes are disseminated to at least one of the plurality of distributed components servicing the distributed applications wherein at least a first application is executed on a first operating system and a second application is executed on a second operating system, where the first and second operating systems are not the same.

Yu teaches that the arbitrator sends out a statistics collection request to all servers. However, the statistics are not configuration change information. Yu also does not teach applications that are executed on two computers with different operating systems. For at least these reasons, Yu does not anticipate claim 40.

Claim 41

Claim 41 recites a method of integrating distributed applications. The method includes sending requests for data-related and messaging-related configuration changes from a first host to a central host. At the first host, translation and messaging configuration changes are implemented.

Yu does not suggest or disclose implementing translation and messaging configuration changes at a first host. For at least this reason, Applicant submits that claim 41 is not anticipated by Yu.

II. Section 103 Rejections

Claims 9-11 and 14-15

Claims 9-11 and 14-15 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Yu in view of U.S. Patent No. 5,606,693 (Nilsen). Applicant respectfully disagrees.

Amended claim 9 recites a method including receiving at a first computer system data translation and messaging configuration information wherein the configuration information is accessed and modified by a user and sent to the first computer system.

Nilsen describes a distributed database management application for maintaining information on a plurality of database servers (col. 2, lines 15-35). The database servers inform a central configurator with the identification and access information to a database server for the data (*id.*). The central configurator maintains information on the number and configuration of the database servers and the status of distributed data logging to each server (*id.*). An operator can change the preferred assignment of the database servers for logging purposes with a controller (col. 4, lines 7-11).

Nilsen fails to suggest or disclose that a first computer system receives data translation and messaging configuration information. The Examiner admits that Yu does not explicitly indicate the step of receiving at the first computer system data translation and messaging configuration information from a configuration information input module wherein configuration information is accessed and modified by a user and sent to a first computer system. For this limitation the Examiner points to Nilsen. However, Nilsen describes an operator that configures servers for logging. Nilsen does not mention receiving a data translation and messaging configuration information from a configuration input module. At most, Nilsen describes receiving preferred assignment of the database servers for logging purposes. The way in which servers are assigned for logging data is not equivalent to data translation and messaging configuration information. For at least this reason, Applicant submits that no *prima facie* case of obviousness has been made with respect to claim 9. Claims 10-11 and 14-15 depend directly from claim 9 and are not obvious for at least the same reason.

Claims 7, 13, 20, 29, 36-39, 42-43 and 45

Claims 7, 13, 20, 29, 36-39, 42-43 and 45 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Yu in view of U.S. Patent Number 5,944,783 (Nieten). Applicant respectfully disagrees.

Claims 42 and 43

Claim 42 recites a method of integrating a plurality of multi-platform applications including providing a plurality of integration modules corresponding to a plurality of multi-platform applications. The plurality of integration modules perform data-related and messaging activities enabling communication among the plurality of multi-platform applications. A central host module is provided that includes a central database of configuration data, where the central host module manages and distributes configuration data that includes instructions for allowing communication among the plurality of multi-platform applications.

Nieten teaches methods for transferring information among software agents operating simultaneously on a digital network (Abstract). A first software agent is created with a knowledge base and placed in a community of agents (*id.*). The first software agent transfers information to the other agents either directly or through a central community server (*id.*). The agents can be written in Java so the agents are platform independent (col. 14, lines 11-18). The Java agents can then work with applications where the type of system upon which the agent will operate is not known ahead of time (*id.*). The agents can communicate with other agents, a community server or with a user (col. 13, lines 61-64).

The Examiner points out that Yu does not mention providing a plurality of integration modules corresponding to a plurality of multi-platform applications. Yu and Nieten also do not teach providing a central host module that distributes configuration data that includes instructions for allowing communication among a plurality of multi-platform applications. Rather, Nieten describes a community server that supports a publish and subscribe system, i.e., the community server passes along information about data products to agents that have profiles containing a production specification for the data products. This publish and subscribe system is different from transferring information down to components where the information allows for

communication among the plurality of multi-platform applications. Further, the agents, as described by Nieten, can communicate with one another, ostensibly without receiving instructions from the central community server that enable the communication. Neiten does not teach how the agents communicate or that the agents require information to enable communication among a plurality of corresponding multi-platform applications. For at least these reasons, claim 42 is not unpatentable in view of Yu and Nieten. Claim 43 depends from claim 42 and is not obvious for at least the same reason as offered for claim 42.

Claim 45

Claim 45 recites a distributed application integration system having a central host means for allocating portions of collective configuration information to a plurality of application hosts where the application hosts communication with a plurality of corresponding multi-platform applications and the plurality of applications hosts implement the portion of the collective information-to enable communication among the plurality of corresponding multi-platform applications.

Nieten discloses Java agents that are platform independent. These agents can communicate with the central community server, other agents or a user. However, Nieten does not suggest or disclose that the central community server manages and distributes configuration data to the integration modules, wherein the configuration data includes instructions for allowing communication among the plurality of multi-platform applications. As noted above, Neiten does not teach how the agents communicate or that the agents require information to enable communication among a plurality of corresponding multi-platform applications. Further, Neiten teaches a publish and subscribe for data products, rather than a central host means for allocating portions of information for applications hosts to implement and enable communication among the corresponding multi-platform applications. For at least these reasons, claim 45 is not unpatentable in view of Yu and Nieten.

Claim 29

Claim 29 depends from claim 28 and recites a distributed, multi-platform application integration system. The system requires a central host, application hosts, application connectors

and multi-platform applications corresponding to the application hosts. The application connectors facilitate communication between the application hosts and the multi-platform applications.

The Examiner argues that it would have been obvious to a person of ordinary skill in the art to use Nieten's teaching to allow the server applications in Yu to better communicate data. The Applicant sees no motivation to combine the teachings of Nieten with the teachings of Yu. Yu does not teach computer systems with multiple platforms. Yu does not suggest any communication complications caused by platform or application differences between the servers and the arbitrator or the servers and the requests. For at least this reason, Applicant submits that the claim is patentable distinct over Yu and Nieten.

Claims 7, 13 and 20

Claim 7 depends from claim 2, claim 13 depends from claim 9 and claim 20 depends from claim 16. Both Yu and Nieten fail to teach or disclose the limitations of the independent claims 2, 9 and 16 and similarly fail to suggest or disclose the limitations of claims 7, 13 and 20. Applicant submits that claims 7, 13 and 20 are therefore allowable.

Claims 36-39

Claims 36-39 depend from claim 28 and include limitation to a plurality of application hosts that include corresponding control brokers where the control brokers are configured to communicate with a central registry system to receive configuration data and a plurality of multi-platform applications corresponding to the plurality of application hosts where the multi-platform applications are configured to communicate via the applications hosts in accordance with the communication data.

Both Yu and Nieten fail to teach or disclose application hosts that have control brokers configured to communicate with a central registry system to receive configuration data and a plurality of multi-platform applications corresponding to the plurality of application hosts where the multi-platform applications are configured to communicate via the applications hosts in accordance with the communication data. For at least this reason, Applicant submits that no *prima facie* case of obviousness has been made with respect to claims 36-39.

Claim 1

Claim 1 was rejected under 35 U.S.C. 103(a) as being unpatentable over Nieten in view of Yu and further in view of Nilsen. Applicant respectfully disagrees.

Claim 1 recites a scalable enterprise application collaboration system. The system requires a central host including a fault tolerant central registry system having a first central registry and a redundant central registry, where the central host is configured to manage a plurality of reusable distributed objects, send configuration change alerts to the plurality of reusable distributed objects and provide configuration data to the plurality of reusable distributed objects from one of the first central registry and the redundant central registry, where if the first central registry is unavailable, the redundant central registry is used.

The Examiner argues that Nilsen teaches a fault tolerant central registry system having a first central registry system and a redundant central registry. Nilsen describes configuration controllers that provide a redundant control system for a network (col. 3, lines 32-36). Nilsen's controllers manage the process through which data is logged from a workstation to database servers (col. 3, lines 44-49). Nilsen's system does not manage a plurality of reusable distributed objects, send configuration change alerts to the plurality of reusable distributed objects and provide configuration data to the plurality of reusable distributed objects from one of the first central registry and the redundant central registry. For at least this reason, the Applicant submits that no *prima facie* case of obviousness has been made with respect to claim 1.

Claims 6, 12, 19 and 44

Claims 6, 12, 19 and 44 were rejected under 35 U.S.C. 103(a) as being unpatentable over Yu in view of U.S. 6,445,919 (Suovieri). Applicant respectfully disagrees.

Claims 6, 12, 19 and 44 are each dependent claims requiring that the configuration information includes data translation and messaging information.

Applicant submits that each of the claims from which claims 6, 12, 19 and 44 depend are not anticipated or obvious in light of the cited prior art, as discussed above. Even if Suovieri describes the additional limitation as required by claims 6, 12, 19 and 44 (which is not conceded) Suovieri fails to suggest or disclose the elements of the independent claims that are not taught, as

described above. For at least this reason, Applicant submits that no *prima facie* case of obviousness has been made and claims 6, 12, 19 and 44 are allowable.

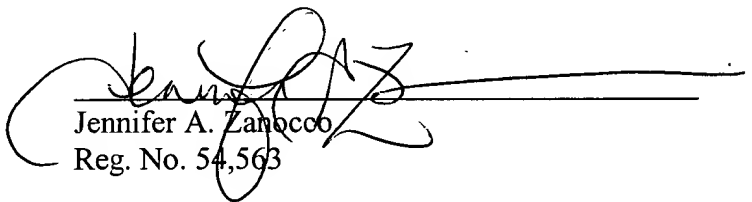
III. New Claims

Claims 46-54 are new claims. No new matter has been added. Claims 46-47 depend from claim 2, claims 48-49 depend from claim 3 and claims 50-51 depend from claim 23. Claims 52, 53 and 54 correspond to claims 41, 40 and 23 respectively. For at least these reasons, Applicant submits that claims 46-54 are allowable as written.

Enclosed are a \$420 check for excess claim fees and a \$420 check for the Petition for Extension of Time fee. Please apply any other charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: May 17, 2004


Jennifer A. Zambocco
Reg. No. 54,563

Fish & Richardson P.C.
500 Arguello Street, Suite 500
Redwood City, California 94063
Telephone: (650) 839-5070
Facsimile: (650) 839-5071